```
%% Inputs:
% Angles: t2v, t3_v, t4v
% Mechanism lengths: a, b, c, d
% End-effector position: AP, tAP
% Support forces: F12, F14
%% Evaluate positions of points O2,A,B,O4,P
%Pegar os dados da atividade 3; FAZER MATRI AX=b
N = 27;
delta = N/4;
d = (100+delta)/1000; %[m]
a = (40-delta)/1000; %[m]
b = (120+delta)/1000; %[m]
c = (80-delta)/1000; %[m]

t2i = 2*pi()/9; %[RAD]
w2 = 4*pi(); %[RAD/s]
a2 = 0; %[RAD/s^2]
dl = 3; %[kg/m]
m2 = a*dl; %[kg]
m3 = b*dl; %[kg]
m4 = c*dl; %[kg]
IG2 = m2*a^2/12; %[kg*m^2] I no centro
IG3 = m3*b^2/12; %[kg*m^2] I no centro
IG4 = m4*c^2/12; %[kg*m^2] I no centro
g = 9.81; %[m/s^2]
% End-effector P
AP=70/1000; tAP=20;
%% Setup of input
% Number of steps
Passos=1000;
% Vector of time instants
t=linspace(0,1,Passos)';
% Vector of values for t2 (for imposed t2 in constant steps)
% t2v=t2i+linspace(0,4*pi,Passos)';
% Vector of values for t2 (for imposed w2 constant)
t2v = t2i+w2.*t;
w2v = w2.*ones(size(t));
a2v = zeros(size(t));
% Vector of values for t2 (for imposed a2 constant)
% a2=8*pi; t2v=t2i+a2/2*t.^2; a2v=a2*ones(size(t)); w2v=a2*t;
%% Newton-Raphson algorithm for the evaluation of t3 and t4 given t2
tol = 1;
% Initial guesses for t3 and t4
t3 = pi()*0.014/180; t4=pi()*40.014/180; % Circuito aberto ou cruzado?
for it2=1:length(t2v)
t2 = t2v(it2); B=tol+1; iconv=0;
while norm(B)>tol
iconv=iconv+1;
A = [-b*sin(t3) c*sin(t4);b*cos(t3) -c*cos(t4)];
B = [a*cos(t2)+b*cos(t3)-c*cos(t4)-d; a*sin(t2)+b*sin(t3)-c*sin(t4)];
Dt = -A\B;
t3 = t3+Dt(1);
t4 = t4+Dt(2);
end
if iconv>2 disp([it2 iconv]), end % Show number of iterations required to converge
```

```
t3_v(it2,1)=t3;
t4v(it2,1)=t4;
end
%% Post-processing for velocities and accelerations
rO2 = zeros(length(t2v),2);
rA = a*[cos(t2v) sin(t2v)];
rB = rA+b*[cos(t3_v) sin(t3_v)];
rO4 = [rO2(:,1)+d rO2(:,2)];
rP = rA+AP*[cos(t3_v+tAP) sin(t3_v+tAP)];

tmqv = t3_v-t4v;
qmtv = t4v-t3_v;
qmdv = t4v-t2v;
dmtv = t2v-t3_v;

w3v = a*w2v.*sin(qmdv)./(b*sin(tmqv));
w4v = a*w2v.*sin(dmtv)./(c*sin(qmtv));

A = c*sin(t4v);
B = b*sin(t3_v);
C = a*a2v.*sin(t2v) + a*w2v.*w2v.*cos(t2v) + b*w3v.*w3v.*cos(t3_v) - c*w4v.*w4v.*cos↙
(t4v);
D = c*cos(t4v);
E = b*cos(t3_v);
F = a*a2v.*cos(t2v) - a*w2v.*w2v.*sin(t2v) - b*w3v.*w3v.*sin(t3_v) + c*w4v.*w4v.*sin↙
(t4v);

a3v = ((C.*D)-(A.*F))./((A.*E)-(B.*D));
a4v = ((C.*E)-(B.*F))./((A.*E)-(B.*D));

vpontoA = a.*w2v;
apontoA = (((a2v.*a).^2)+((w2v.^2).*a).^2).^(0.5); %Apenas aceleração centrípeta
apontoCG2_x = ((a2v.*a/2).*sin(t2v))-((w2v.^2).*a/2).*cos(t2v);
apontoCG2_y = ((a2v.*a/2).*cos(t2v))-((w2v.^2).*a/2).*sin(t2v);
apontoCG4_x = -((a4v.*c/2).*sin(t4v))-((w4v.^2).*c/2).*cos(t4v);
apontoCG4_y = ((a4v.*c/2).*cos(t4v))-((w4v.^2).*c/2).*sin(t4v);
apontoCG3_x = -apontoA.*cos(t2v)-(b/2).*a3v.*sin(t3_v)-(b/2).*w3v.^2.*cos(t3_v);
apontoCG3_y = -apontoA.*sin(t2v)+(b/2).*a3v.*cos(t3_v)-(b/2).*w3v.^2.*sin(t3_v);

T12 = [];
F12 = [];
F14 = [];

for i=1:length(t2v)
Var1 = (a/2)*sin(t2v(i));
Var2 = (a/2)*cos(t2v(i));
Var3 = (b/2)*sin(t3_v(i));
Var4 = (b/2)*cos(t3_v(i));
Var5 = (c/2)*sin(t4v(i));
Var6 =(c/2)*cos(t4v(i));
I = [[0 1 0 1 0 0 0 0 0];
[0 0 1 0 1 0 0 0 0];
[1 (Var1) (-Var2) (-Var1) (Var2) 0 0 0 0];
[0 0 0 -1 0 0 0 -1 0];
[0 0 0 0 -1 0 0 0 -1];
[0 0 0 -Var3 Var4 0 0 Var3 -Var4];
[0 0 0 0 0 1 0 1 0];
[0 0 0 0 0 0 1 0 1];
[0 0 0 0 0 Var5 -Var6 -Var5 Var6]];
K = [m2*apontoCG2_x(i);
    g*m2+m2*apontoCG2_y(i);
    0;
    m3*apontoCG3_x(i);
    m3*apontoCG3_y(i)+m3*g;
    IG3*a3v(i);
    m4*apontoCG4_x(i);
    m4*apontoCG4_y(i)+m4*g;
    IG4*a4v(i)];
J = I\K;

T12 = [T12, J(1)];
F12 = [F12, sqrt((J(2))^2+(J(3))^2)];
F14 = [F14, sqrt((J(6))^2+(J(7))^2)];

end;

figure(2);
img2=plot(t, T12, '-b');
grid on;
xlabel('Tempo (s)'); ylabel('Torque T12 (N.m)');
hold on;

figure(3);
img3=plot(t, F12, '-g');
grid on;
xlabel('Tempo (s)'); ylabel('Força F12 (N)');
hold on;

figure(4);
img4=plot(t, F14, '-r');
grid on;
xlabel('Tempo (s)'); ylabel('Força F14 (N)');
hold on;
max(F12)
min(F12)
max(F14)
min(F14)
max(T12)
min(T12)
%% Show simulation
outvid=0; % if =1 outputs GIF video animation
figure(1), clf, set(1,'position',[0 0 690 650])
if outvid==1
filename='qbarras_nr.gif';
frame=getframe(gcf); im=frame2im(frame); [A,map]=rgb2ind(im,256);
imwrite(A,map,filename,'gif','LoopCount',Inf,'DelayTime',1e-3);
end
rT=[rO2; rA; rB; rO4; rP]; mx=max(max(abs(rT)));
rT=[rO2; rA; rB; rO4]; mx=max(max(abs(rT))); axis([-mx mx -mx mx]),
axis tight, axis equal, axis off,
hAP=line([rA(1,1) rP(1,1)],[rA(1,2) rP(1,2)]); set↙
(hAP,'Color','g','LineStyle','-','Marker','o'),
hPB=line([rP(1,1) rB(1,1)],[rP(1,2) rB(1,2)]); set↙
(hPB,'Color','g','LineStyle','-','Marker','o'),
hO2A=line([rO2(1,1) rA(1,1)],[rO2(1,2) rA(1,2)]);
set(hO2A,'Color','k','LineStyle','-','Marker','o'),
hAB=line([rA(1,1) rB(1,1)],[rA(1,2) rB(1,2)]); set↙
(hAB,'Color','k','LineStyle','-','Marker','o'),
hBO4=line([rB(1,1) rO4(1,1)],[rB(1,2) rO4(1,2)]);
set(hBO4,'Color','k','LineStyle','-','Marker','o'),
hO4O2=line([rO4(1,1) rO2(1,1)],[rO4(1,2) rO2(1,2)]);
set(hO4O2,'Color','k','LineStyle','-','Marker','o'),
hO2Ao=line([rO2(1,1) rA(1,1)],[rO2(1,2) rA(1,2)]); set(hO2Ao,'Color',.7*[1 1↙
1],'LineStyle',':'),
hABo=line([rA(1,1) rB(1,1)],[rA(1,2) rB(1,2)]); set(hABo,'Color',.7*[1 1↙
1],'LineStyle',':'),
hBO4o=line([rB(1,1) rO4(1,1)],[rB(1,2) rO4(1,2)]); set(hBO4o,'Color',.7*[1 1
1],'LineStyle',':'),
hO4O2o=line([rO4(1,1) rO2(1,1)],[rO4(1,2) rO2(1,2)]); set(hO4O2o,'Color',.7*[1 1
1],'LineStyle',':'),
text(rO2(1,1)-mx/100,rO2(1,2)-mx/20,'$O_2$')
text(rA(1,1)-mx/100,rA(1,2)+mx/20,'$A$')
text(rB(1,1)-mx/100,rB(1,2)+mx/20,'$B$')
text(rO4(1,1)-mx/100,rO4(1,2)-mx/20,'$O_4$')
text(rP(1,1)-mx/100,rP(1,2)+mx/20,'$P$')
% Support forces and torques
maxF=max(max(abs([F12 F14])))/(a/2); Fs=F12+F14;
hq12=line([rO2(1,1) rO2(1,1)-F12(1,1)/maxF],[rO2(1,2) rO2(1,2)-F12(1,2)/maxF]);
hq14=line([rO4(1,1) rO4(1,1)-F14(1,1)/maxF],[rO4(1,2) rO4(1,2)-F14(1,2)/maxF]);
hqs=line([(rO2(1,1)+rO4(1,1))/2
(rO2(1,1)+rO4(1,1))/2-Fs(1,1)/maxF],[(rO2(1,2)+rO4(1,2))/2
(rO2(1,2)+rO4(1,2))/2-Fs(1,2)/maxF]);
if outvid==1 dts=5; else dts=1; end
for n=2:dts:length(t)
set(hO2A,'xdata',[rO2(n,1) rA(n,1)],'ydata',[rO2(n,2) rA(n,2)]);
set(hAB,'xdata',[rA(n,1) rB(n,1)],'ydata',[rA(n,2) rB(n,2)]);
set(hBO4,'xdata',[rB(n,1) rO4(n,1)],'ydata',[rB(n,2) rO4(n,2)]);
set(hAP,'xdata',[rA(n,1) rP(n,1)],'ydata',[rA(n,2) rP(n,2)]);
set(hPB,'xdata',[rP(n,1) rB(n,1)],'ydata',[rP(n,2) rB(n,2)]);
hA=line([rA(n-1,1) rA(n,1)],[rA(n-1,2) rA(n,2)]); set(hA,'Color','r','LineStyle',':');
hB=line([rB(n-1,1) rB(n,1)],[rB(n-1,2) rB(n,2)]); set(hB,'Color','r','LineStyle',':');
hP=line([rP(n-1,1) rP(n,1)],[rP(n-1,2) rP(n,2)]); set(hP,'Color','m','LineStyle',':');
set(hq12,'xdata',[rO2(n,1) rO2(n,1)-F12(n,1)/maxF],'ydata',[rO2(n,2) rO2(n,2)-F12(n,2)↙
/maxF]);
set(hq14,'xdata',[rO4(n,1) rO4(n,1)-F14(n,1)/maxF],'ydata',[rO4(n,2) rO4(n,2)-F14(n,2)↙
/maxF]);
set(hqs,'xdata',[(rO2(n,1)+rO4(n,1))/2 (rO2(n,1)+rO4(n,1))/2-Fs(n,1)/maxF],'ydata',↙
[(rO2(n,2)+rO4(n,2))/2 (rO2(n,2)+rO4(n,2))/2-Fs(n,2)/maxF]);
if outvid==1
    frame=getframe(gcf); im=frame2im(frame); [A,map]=rgb2ind(im,256);
    imwrite(A,map,filename,'gif','WriteMode','append','DelayTime',1e-3);
else pause(1e-12);
end
end
```